

6、ZigBee 网关 485 通信

1.实验目的

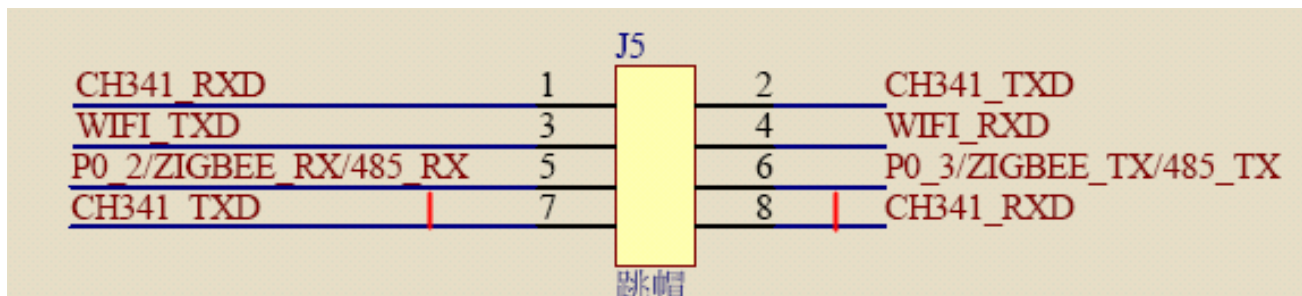
- 1)、通过实验掌握 CC2530 芯片串口配置与使用
- 2)、观察 D2 串口发送指示灯的变化，每发送一串字符闪一次
- 3)、掌握 485 电路运用，注意 485 大量用于工业领域

2.实验设备

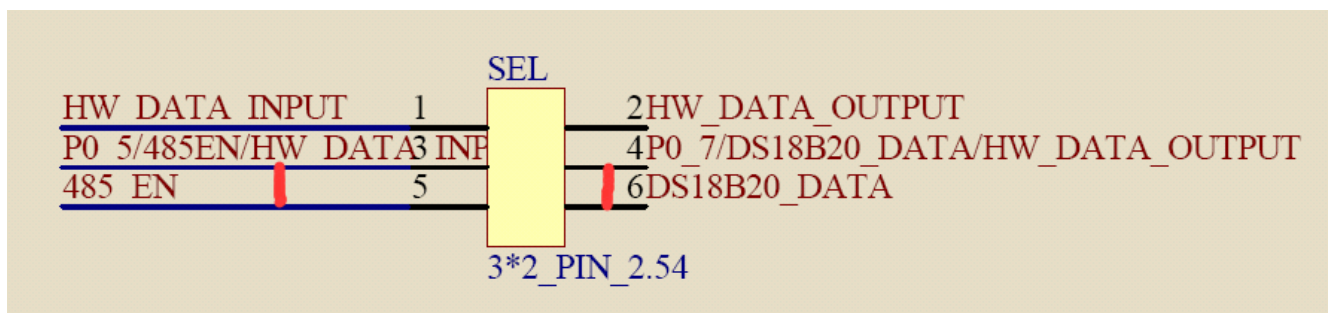
硬件：PC 机一台 ZB 网关（底板、核心板、仿真器、USB 线）一套

软件：2000/XP/win7 系统，IAR 8.10 集成开发环境、串口助手

3.相关电路图

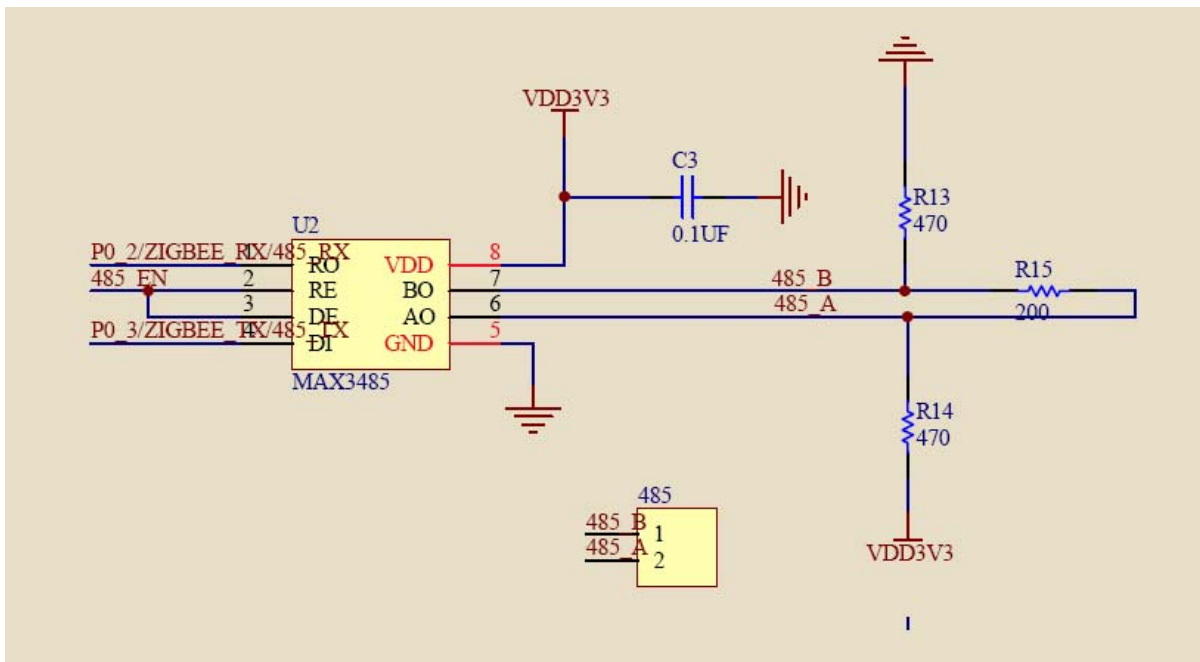


如上图 J5 这样接，表明 PC 通过 USB 线[板子自带 USB 转串口芯片 CH341]直接接入 ZB 模块，进行对 ZB 的调试使用。如果 3-5 4-6 则表明 ZIGBEE 通过串口控制 wifi 模块。如果 1-3 2-4 则表明 WIFI 模块接入到 PC

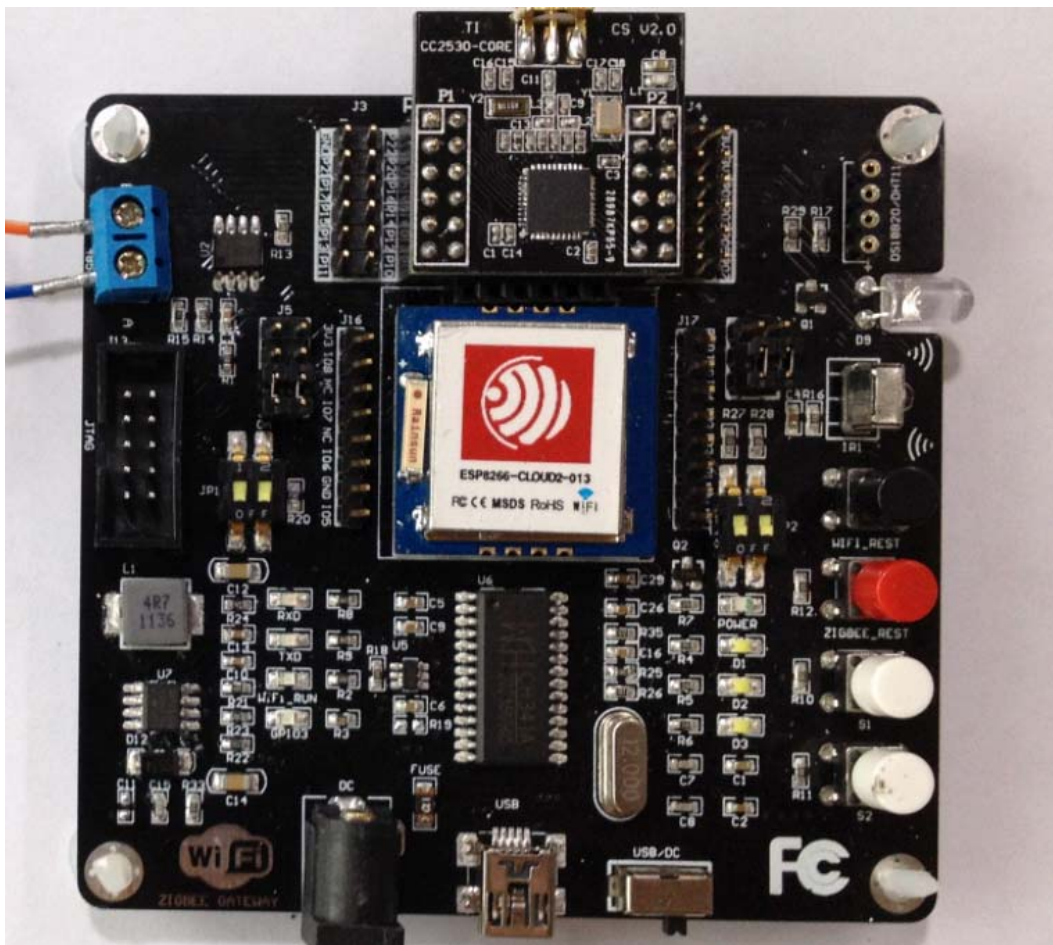


如上图 SEL 这样接，1- 3 2-4。 则表明红外的收发引脚接入到 ZIGBEE 的芯片 IO 进行驱动，其中 HW_DATA_OUTPUT 是红外发射模块。如果 3-5 4-6 表明温度、温湿度和 485 模块直接接入到 ZIGBEE 的芯片 IO 进行驱动，现在就是这样的情况。

如下图是 485 电路图。我们用 max3485 芯片



如下图就是我们板子实际的短路帽情况：





硬件我们准备好了。我们将板子和电脑连接。这个时候需要安装驱动。具体安装方法很简单，下载驱动精灵自动安装。不要自己安装，自己安装容易把驱动搞乱。驱动精灵网址：
<http://pan.baidu.com/s/14fUyU>

P0_2、P0_3 配置为外设功能时：P0_2 为 RX，P0_3 为 TX。USART0 和 USART1 是串行通信接口，它们能够分别运行于异步 UART 模式或者同步 SPI 模式。两个 USART 具有同样的功能，可以设置在单独的 I/O 引脚。此种串口设计是没有流控功能的。

4.相关寄存器

相关寄存器 UxCSR、UxCSR、UxGCR、UxBUF、UxBAUD、CLKCONCMD、CLKCONSTA 如

寄存 器	位	描述
UOCSR (0x86) -USART0 控制 和状态	Bit[7] MODE	USART 模式选择 0: SPI 模式 1: UART 模式
	Bit[6] RE	UART 接收器使能 0: 禁用接收器 1: 接收器使能
	Bit[5] SLAVE	SP 主或者从模式选择 0: SPI 主模式 1: SPI 从模式
	Bit[4] FE	UART 帧错误状态 0: 无帧错误检测 1: 字节收到不正确停止位级别
	Bit[3] ERR	UART 奇偶错误状态 0: 无奇偶错误检测 1: 字节收到奇偶错误
	Bit[2] RX_BYTE	接收字节状态 0: 没有收到字节 1: 准备好接收字节
	Bit[1] TX_BYTE	传送字节状态 0 字节没有被传送 1 写到数据缓存寄存器的最后字节被传送
	Bit[0] ACTIVE	USART 传送/接收主动状态、在 SPI 从模式下 该位等于从模式选择 0: USART 空闲 1: 在传送或者接收模式 USART 忙碌
UOOCR (0xC5) -USART0 通用 控制	Bit[7] CPOL	SPI 的时钟极性 0: 负时钟极性 1: 正时钟极性
	Bit[6] CPHA	SPI 时钟相位 0: 当 SCK 从 CPOL 倒置到 CPOL 时数据输出 到 MOSI, 并且当 SCK 从 CPOL 倒置到 CPOL 时 数据输入抽样到 MISO。 1: 当 SCK 从 CPOL 倒置到 CPOL 时数据输出 到 MOSI, 并且当 SCK 从 CPOL 倒置到 CPOL 时 数据输入抽样到 MISO

	Bit[5] ORDER	传送位顺序 0: LSB 先传送 1: MSB 先传送
	Bit[4:0] BAUD_E	波特率指数值。BAUD_E 和 BAUD_M 决定了 UART 波特率和 SPI 的主 SCK 时钟频率
U0BAUD (0xC2) - USART 0 波特率控制	BAUD_M[7:0]	波特率小数部分的值。BAUD_E 和 BAUD_M 决定了 UART 的波特率和 SPI 的主 SCK 时钟频率
U0DBUF		USART 0 接收/发送数据缓存
UTX0IF(发送中断标志)	IRCON2 Bit1	USART 0 TX 中断标志 0: 无中断未决 1: 中断未决
CLKCONCMD 时钟控制命令	Bit[7] OSC32K	32 kHz 时钟振荡器选择 0 : 32 kHz XOSC 1 : 32 kHz RCOSC
	Bit[6] OSC	系统时钟源选择 0 : 32 MHz XOSC1 : 16 MHz RCOSC
	Bit[5:3] TICKSPD	定时器标记输出设置 000 : 32 MHz 001 : 16 MHz 010 : 8 MHz 011 : 4 MHz 100 : 2 MHz 101 : 1 MHz 110 : 500 kHz 111 : 250 kHz
	Bit[2:0] CLKSPD	时钟速度 000 : 32 MHz 001 : 16 MHz 010 : 8 MHz 011 : 4 MHz 100 : 2 MHz 101 : 1 MHz 110 : 500 kHz 111 : 250 kHz
CLKCONSTA		CLKCONSTA 寄存器是一个只读寄存器, 用来获得当前时钟状态

由寄存器UxBAUD.BAUD_M[7:0]和UxGCR.BAUD_E[4:0]定义波特率。该波特率用于UART 传送，也用于SPI 传送的串行时钟速率。波特率由下式给出：

$$\text{波特率} = \frac{(256 + \text{BAUD_M}) * 2^{\text{BAUD_E}}}{2^{28}} * F$$

F 是系统时钟频率，等于 16 MHz RCOSC 或者 32 MHz XOSC。

32 MHz 系统时钟常用的波特率设置

波特率 (bps)	UxBAUD. BAUD_M	UxGCR. BAUD_E	误差 (%)
2400	59	6	0.14
4800	59	7	0.14
9600	59	8	0.14
14400	216	8	0.03
19200	59	9	0.14
28800	216	9	0.03
38400	59	10	0.14
57600	216	10	0.03
76800	59	11	0.14
115200	216	11	0.03
230400	216	12	0.03

CC2530 配置串口的一般步骤：

- 1、配置 IO，使用外部设备功能。此处配置 P0_2 和 P0_3 用作串口 UART0
- 2、配置相应串口的控制和状态寄存器。
- 3、配置串口工作的波特率。

5.源码分析

```
#include <ioCC2530.h>
```

```
#include <string.h>
```

```
#define uint unsigned int
```

```
#define uchar unsigned char
```

```
//定义控制灯的端口
```



```
#define LED1 P1_0
```

```
#define LED2 P1_1
```

```
#define LED3 P0_4
```

```
//函数声明
```

```
void Delay(uint);
```

```
void initUARTSEND(void);
```

```
void UartTX_Send_String(char *Data,int len);
```

```
char Txdata[25]="HELLO! zigbee!\r\n";
```

```
/******
```

```
延时函数
```

```
***** */
```

```
void Delay(uint n)
```

```
{
```

```
    uint i;
```

```
    for(i=0;i<n;i++);
```

```
    for(i=0;i<n;i++);
```

```
    for(i=0;i<n;i++);
```

```
    for(i=0;i<n;i++);
```

```
    for(i=0;i<n;i++);
```

```
}
```

```
/******
```

```
串口初始化函数
```

```
***** */
```

```
void initUARTSEND(void)
```

```
{
```

```
    CLKCONCMD &= ~0x40;
```

```
    while(CLKCONSTA & 0x40);
```

```
    CLKCONCMD &= ~0x47;
```

```
//设置系统时钟源为 32MHZ 晶振
```

```
//等待晶振稳定
```

```
//设置系统主时钟频率为 32MHZ
```

```

PERCFG = 0x00;           //位置 1 P0 口
POSEL = 0x0c;           //P0_2,P0_3 用作串口
P2DIR &= ~0XC0;         //P0 优先作为 UART0

U0CSR |= 0x80;           //UART 方式
U0GCR |= 11;            //U0GCR.BAUD_E
U0BAUD |= 216;           //波特率设为 115200 UxBAUD.BAUD_M
UTX0IF = 0;             //UART0 TX 中断标志初始置位 0

```

```

}

```

```

/*****

```

串口发送字符串函数

```

*****/

```

```

void UartTX_Send_String(char *Data,int len)

```

```

{
    int j;
    for(j=0;j<len;j++)
    {
        U0DBUF = *Data++;
        while(UTX0IF == 0);
        UTX0IF = 0;
    }
}

```

```

}

```

```

/*****

```

主函数

```

*****/

```

```

void main(void)

```

```

{
    uchar i;
    P1DIR = 0x03;         //P1 控制 LED
    initUARTSEND();
}

```



```

LED1 = 1;                //关 LED
LED3 = 1;                //关 LED
LED2 = 0;
UartTX_Send_String(Txdata,25);
for(i=0;i<30;i++)Txdata[i]=' ';
strcpy(Txdata,"HELLO! zigbee!\r\n");    //将 UART0 TX test 赋给 Txdata;
while(1)
{
    UartTX_Send_String(Txdata,strlen(Txdata)); //串口发送数据
    Delay(50000);                //延时
    LED2=!LED2;                //D2 灯在闪，标志发送状态
    Delay(50000);
    Delay(50000);
}
}

```

6、实验现象

把程序下到开发板上，同时把【232-485 转换器

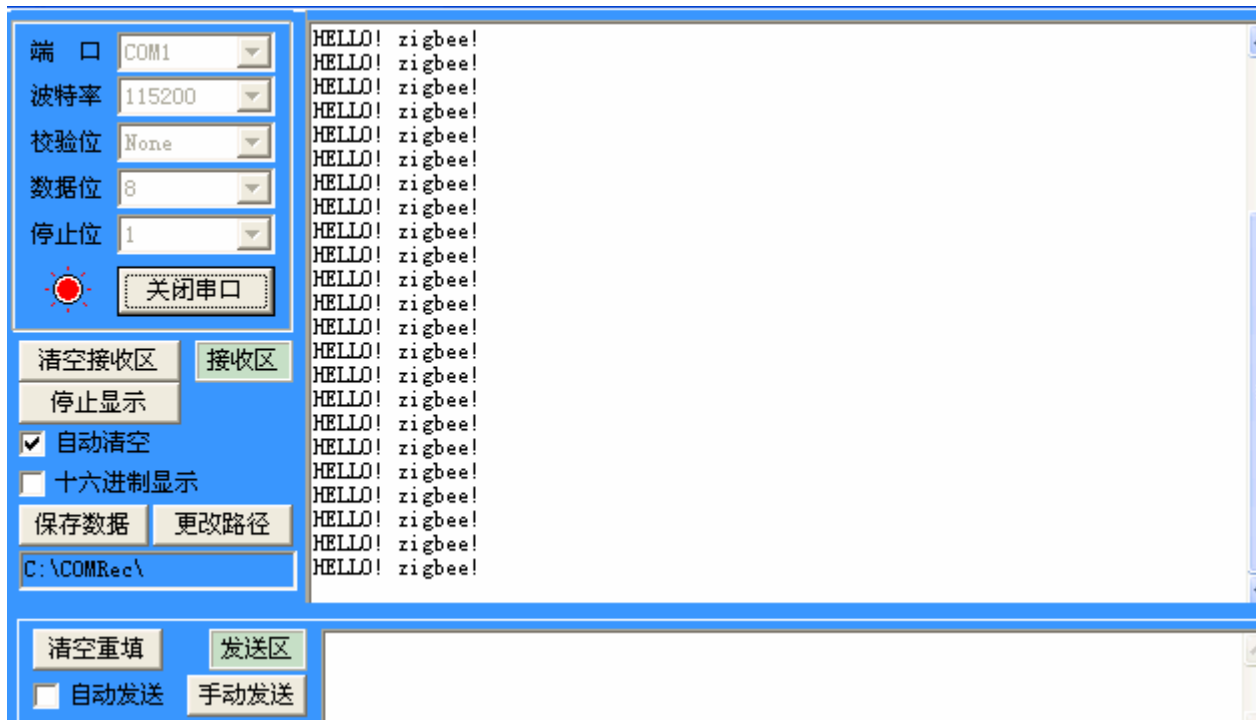


】线接到 PC 上。PC 上面有自带

的串口 COM1 或者自己备【USB 串口线

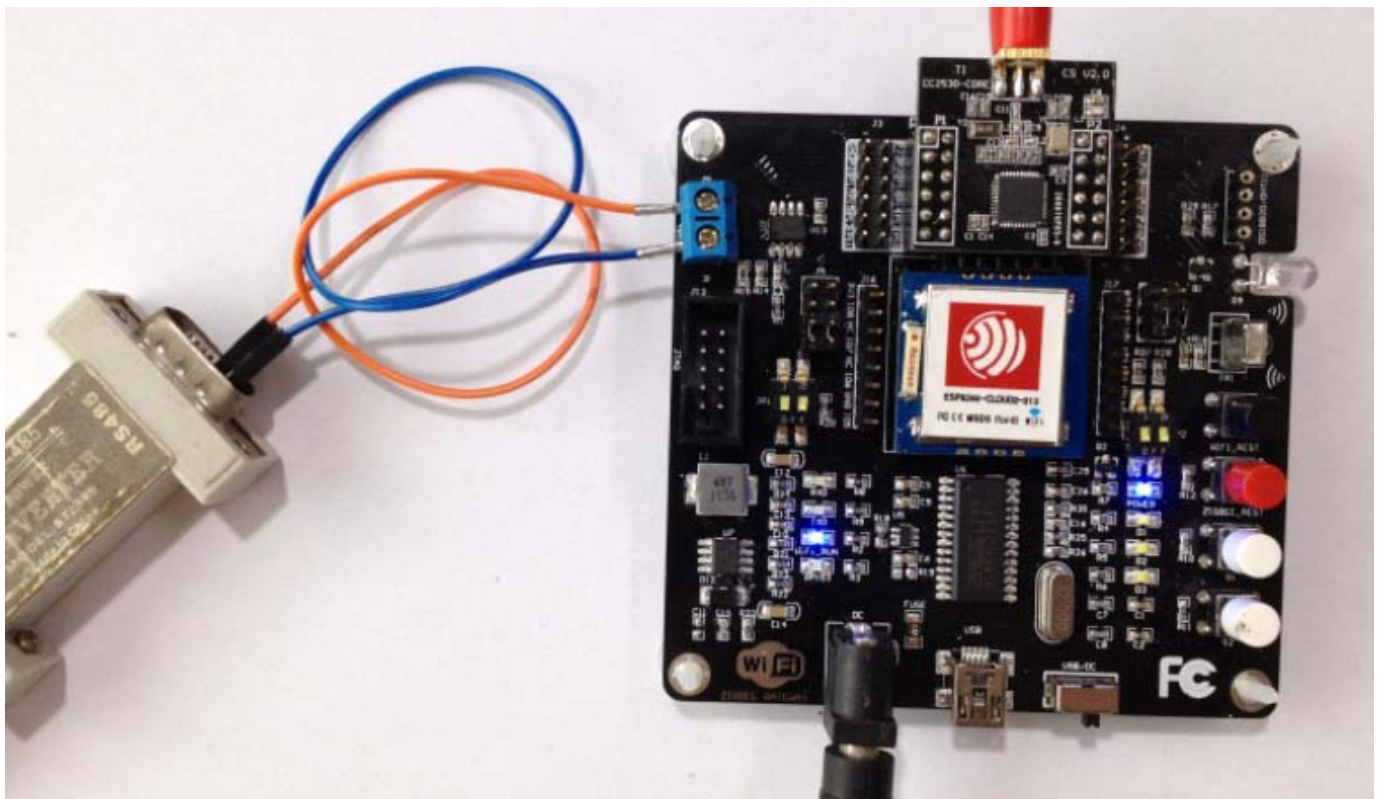


然后把板子的 485->A---[232-485 转换器]的 1 脚、485->B----[232-485 转换器]的 2 脚
如下图：



上图：COM 或者自带 USB 串口线 测试效果！

实物如下图：



注意：如果没收到，检查下跳线帽是否是对的！还可以关闭串口软件，重新插拔 USB 线。